

Write a program to calculate the square root of a number in Python

In this article, TipsMake.com will help you learn more about square roots and how to write a program to calculate square roots in Python programming language.

Problem : Given a number, write a program to find the square root of that number.

Example :

Input: 25 Output: 5

In this article, TipsMake.com will help you learn more about square roots and how to write a program to calculate square roots in Python programming language.

Square root in math

In algebra, the square of a number is the result of the number multiplying by itself: $x = n*n$, where $n*n = n^2$.

You can calculate the square in Python like this:

```
>>> n = 5 >>> x = n ** 2 >>> x 25
```

The `**` operator in Python is used to calculate the square of a number. In this case, 5 squared or 5 times 5 equals 25.

Then, the square root is the number n that, when multiplied by itself, gives the square of x . In this example, the square root is 5.

25 is a perfect square. Perfect squares are squares of integer values.

Perfect square example:

```
>>> 1 ** 1 1 >>> 2 ** 2 4 >>> 3 ** 3 9
```

You may have memorized these perfect squares when you were studying the times table. If you are given a perfect square of a small value, you can easily calculate or memorize its square root. However, for other squares, this calculation can be a bit more complicated.

But as a programmer, you can create your own square root calculator.

Write a program to calculate square root in Python

Python's math module, a standard library, can help you solve math problems while coding. It contains a lot of useful functions like remainder() and factorial(). It also has a function dedicated to calculating square roots, sqrt().

You'll start by declaring the math module:

```
>>> import math
```

Now, you can use the math.sqrt() function to calculate the square root of a number.

sqrt() has a pretty straightforward interface. It takes only one parameter x, as you can see in the examples above, x represents the square you're trying to find the square root of.

The return value of sqrt() is the square root of x, as a floating point. If x = 25 as in the above example, the return result will be 5.0.

We will look at some concrete examples and ways to find the square root of a number using sqrt() and without using sqrt().

Square root of a positive number

Positive numbers are the types of arguments you can pass to the sqrt() function, including int and float types.

Example: You can calculate the square root of 49 using the sqrt() function simply like this (remember to declare the math module first):

```
>>> math.sqrt(49) 7.0
```

The returned result will be 7.0 as a floating point number.

Besides integers, you can also include float values in the sqrt() function:

```
>>> math.sqrt(70.5) 8.396427811873332
```

You can check the accuracy of the square root calculation by performing the squared calculation:

```
>>> 8.396427811873332 ** 2 70.5
```

Calculate the square root of the number 0

Even zero can be included in Python's square root function:

```
>>> math.sqrt(0) 0.0
```

Although you don't need to calculate the square root of 0 because you already know the result, we still need to mention it because otherwise there will be times when you put a variable into the sqrt() function and its value is not true. knowledge.

Calculate the square root of a negative number

The square of all real numbers cannot be negative. The result of a multiplication between two numbers can only be negative if a negative number is multiplied by a positive number. By definition, the square of a number is the result of multiplying it by itself that measure cannot have a negative squared number.

```
>>> math.sqrt(-25) Traceback (most recent call last): File "", line 1, in ValueE
```

If you try to give the `sqrt()` function a negative number, you will get a `ValueError` error message because the function is not programmed to calculate the square root of negative numbers. Therefore, calculating the square root of a negative number requires a more complex algorithm, which is outside the scope of Python's square root function.

If we are not allowed to use the `sqrt()` function, what can we do?

In fact, when you attend interviews for Python programmer positions, you may be asked to write a program to calculate the square root of a number, but you are not allowed to use Python's built-in functions. Therefore, now you will have to build your own algorithm to solve this problem.

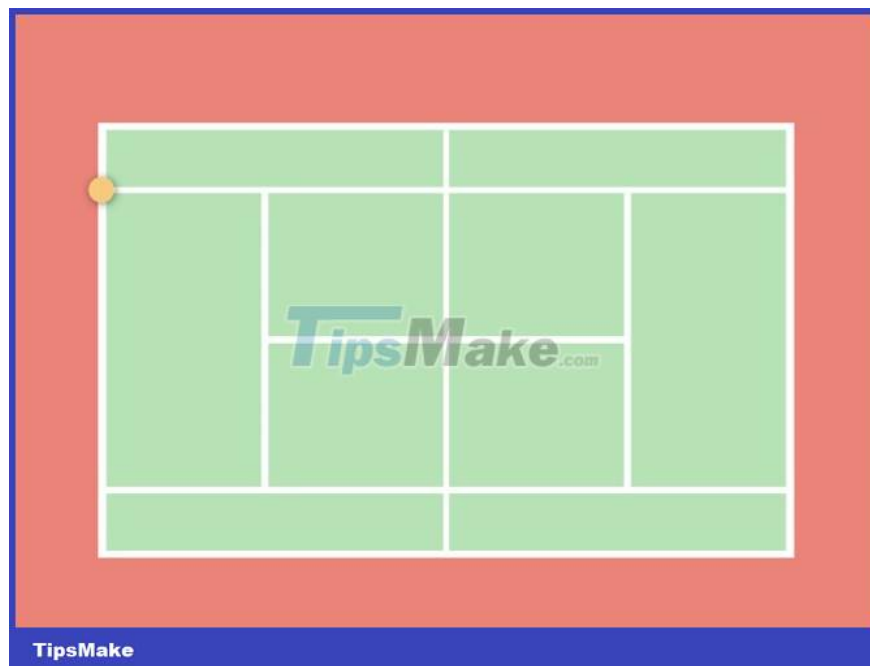
Here is the sample code of this case for your reference:

```
def mySqrt(x): left = 1 right = x mid = 0 while (left = right): mid = (left + ri
```

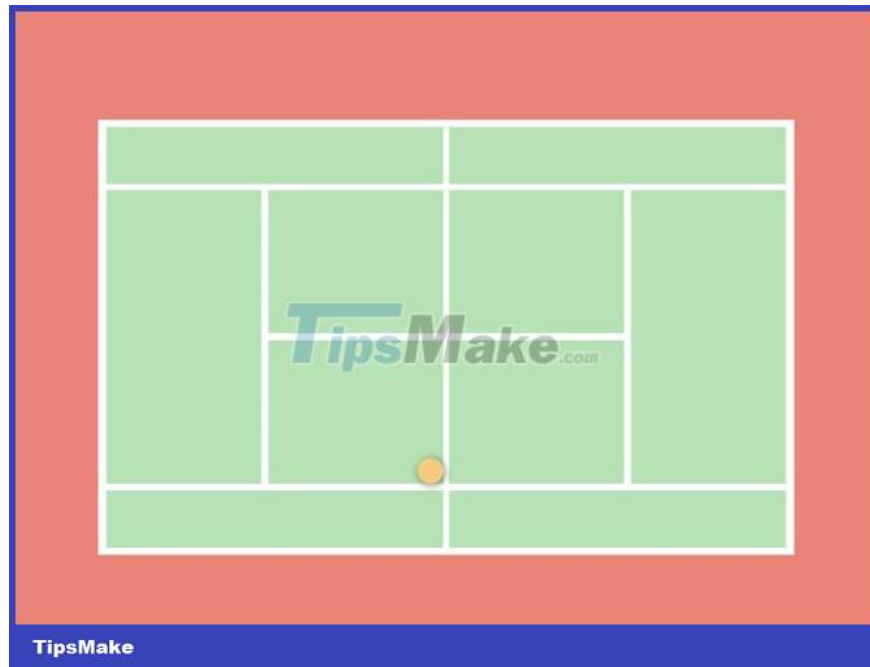
Applications of square roots in real life

To see the application of a square root calculator in real life, we will consider the following example:

Imagine that Rafael Nadal, one of the fastest tennis players in the world, just hit a forehand in the back corner, where the bottom line meets the court line.

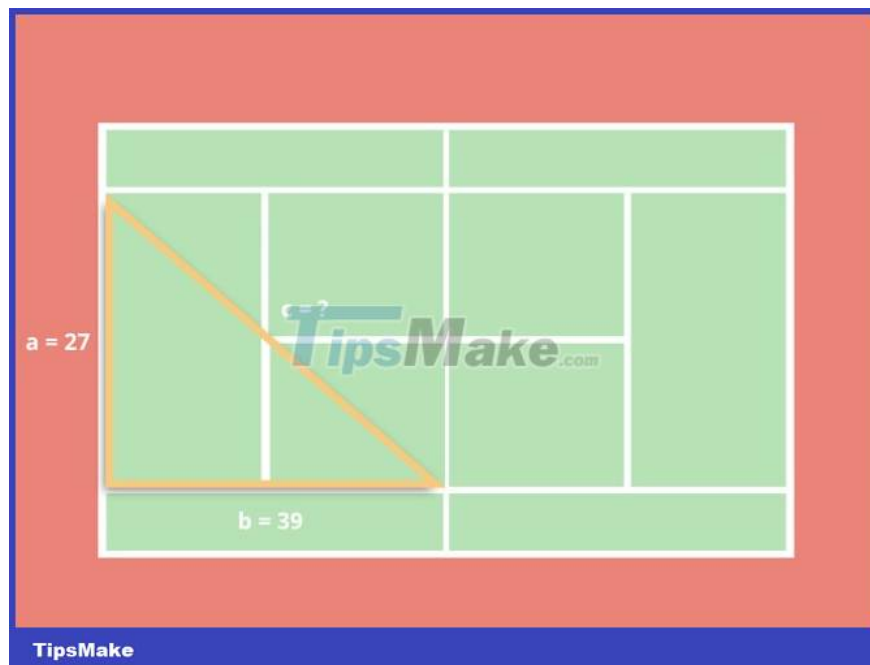


Now, suppose Nadal's opponent hits back with a hit into the net to the opposite corner, the other side corner touching the foot of the net.



How far does Naldal have to run to catch the ball?

You can calculate based on the prescribed size of a tennis court with a 27-foot (8.23-meter) end line and 39 feet (11.88 meters) long (half-court) sideline. So this is basically a problem of finding the length of the hypotenuse of a right triangle.



We know that for a right triangle $a^2 + b^2 = c^2$. Thus, we can rearrange the equation to find c:

Picture 4 of Write a program to calculate the square root of a number in Python

Now, we can solve this equation using Python's square root function:

```
>>> a = 27 >>> b = 39 >>> math.sqrt(a ** 2 + b ** 2) 47.43416490252569
```

From there, the result is that Nadal has to run about 47.4 feet (14.5 meters) to be able to reach the ball and keep the point.

TipsMake.com hopes that this article will be useful to you.

You finished reading the article "**Write a program to calculate the square root of a number in Python**" edited by the [TipsMake](#) team. We hope this article has provided you with many useful tech tips and tricks. You can search for similar articles on tips and guides. Thank you for reading and for following us regularly.